



U11.7 Project: find the one that is green

Vision · University · about 50 min

Name _____

Class _____

Date _____

What this lesson is about

Three identical shapes, one job, and a task the depth sensor cannot begin.

- The job
- A shape to start from
- What is worth thinking about
- What comes next

Questions

6 marks in all

1. Three identical balls differ only in colour. Why can the depth sensor not pick out the green one? [1 mark]

- ☐ A It measures a property of space, and identity lives in the surfaces, which only the camera sees
- ☐ B Its range is too short to reach the balls
- ☐ C Its readings are too noisy
- ☐ D It cannot measure bearings

2. The survey finds three 6 cm balls. What does this print? [1 mark]

```
import math
F, R = 92.4, 3.0
seen = {"red": (100, 14), "green": (200, 28), "blue": (250, 9)}
for colour, (cx, width) in seen.items():
    got = 2 * R * F / width, math.degrees(math.atan((cx - 160) / F))
    print(colour, "%.0f cm at %.0f degrees" % got)
```

3. Why turn to the green ball's bearing and servo on its apparent size, rather than work out its coordinates [1 mark] and drive there open loop?

- ☐ A The range is far worse than the bearing, and closing the loop on size lets the measurement improve as the robot gets closer
- ☐ B Coordinates cannot be computed from a bearing and a range
- ☐ C Open loop driving is not allowed on the mat
- ☐ D Servoing is faster than driving a straight line

4. Why must the survey be a separate phase before the approach? [1 mark]

- ☐ A The camera runs one detector at a time, so it cannot watch for red while servoing on green
- ☐ B The balls move once the robot sets off
- ☐ C The depth sensor must be switched off while the camera runs
- ☐ D The bearings change units when the robot moves

5. The approach should stop 20 cm from a 6 cm ball. About how many pixels wide should it look, to 1 decimal place, with $f = 92.4$? [1 mark]

6. Which choices help the robot stop short of the ball without touching it? [1 mark]

Tick every answer that is true.

- ☐ A Use the dead band trick rather than a large forward gain
- ☐ B Stop on a band of widths rather than one exact width
- ☐ C Raise the forward gain so it settles before it can overshoot
- ☐ D Plan the approach from the survey range and drive it without looking

The task: find the one that is green

Survey all three balls and print `red:`, `green:` and `blue:`, the range in centimetres to each. Then drive to the green ball and stop within reach of it without touching it. `position()` is not allowed: the robot has to find it by looking.

```
from bugbot import *
import math
connect()
```

```
F, R = 92.4, 3.0
TARGET = "green"
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/u11-7-project-find-the-green/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Run the survey again from where you finished. Which ranges improved, and by how much compared with the d^2 prediction?
2. Make the choice of colour come from a tag: read a marker first and let its id decide which ball to fetch. What has to change in the program, and what does not?
3. Replace the colour test with size: fetch the largest ball rather than the green one. Which measurement is doing the work now, and which of the two would you rather rely on in a room with a window?

